

**REMARKS**

Claims 1-3 and 6-12 were rejected in the parent case under 35 U.S.C. § 102(e) as being anticipated by Garrity et al., U.S. Patent No. 6,230,205. Claims 4-5 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Garrity et al., further in view of Dobbins et al., Published Application 2002/0066033. These rejections are traversed for the reasons previously advanced in the response to the First Office Action, which was submitted on or about June 24, 2004. Among other differences, Garrity et al. do not teach a content delivery network, let alone any hierarchical mechanism for purging previously stored data from a content server in such a network. Further, Garrity et al. do not teach the “push-pull” aspects of the invention, e.g., use of a set of staging servers that receive aggregate purge requests pushed to them, where a given content server then obtains the request (in effect, pulls it) from a staging server.

MPEP § 2131 provides that a “claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. ... ‘The identical invention must be shown in as complete detail as contained in the ... claim.’ The elements must be arranged as required by the claim.” (citations omitted, emphasis supplied). For at least the above reasons, the Examiner is incorrect that all elements or steps of each of claims 1 and 11 are disclosed in Garrity et al. They are not. Thus, the anticipation rejection should be withdrawn.

To advance this prosecution, independent claims 1 and 11 have been further amended to more clearly distinguish this invention. In particular, the preamble of each claim now clarifies that the invention is practiced in a network in which the content servers (from which content files are to be purged) also share content with each other. In this situation, a first content server may have executed a purge request while a second content server (with whom the first content server shares content) may not have executed the purge request. This can lead to circumstances in which the second content server attempts to provide a given content file to the first content server although the first content server has purged that content already. A discussion of this problem and the solution provided begins on page 31 of the written description, in Section 12.3.2. Each of the independent claims has been amended to include this functionality. Thus, for example, independent claim 11 states the given content server also includes code for inhibiting given data sharing between the given content server and at least one other content server with which the

given content server shares content files if the other content server has not then removed a given content file that has been removed by the given content server. The method described in independent claim 1 describes a similar functionality, namely, inhibiting the first content server from receiving, from a second content server in the set of content servers with which the first content server shares content files, a given content file in the aggregate purge request if the second content server has not then executed the aggregate purge request.

Neither cited reference nor any other art of record remotely discloses or suggests such a function. Garrity et al. do not even describe a content delivery network, and the machines in that system are not content servers “that share content files with each other.” As a consequence, the Garrity et al. machines do not have the synchronization problem addressed by the modified claim language, namely, a situation where servers share content but receive and/or act upon purge instructions at different times, thereby creating the possibility of having to deal with a content file that has been purged from one machine but not the other.

In the parent case final rejection at paragraph 10, the Examiner cited two portions of the Garrity et al. specification (“authorization of file transfer, col 15, lines 7-13” and “the server processes/threads are appropriately synchronized to any share data, Col 28, lines 27-47”) to support the argument that dependent claim 12 was anticipated. The first citation relating to “authorization of file transfer” is not understood as this is not what applicants were describing in claim 12 (or amended claims 1 or 11). As to the second point, the Examiner’s citation is taken out of context, as the actual text reads as follows (with emphasis supplied): “In the depicted example, a single server process, which contains multiple threads, services all clients to server 400. In such a situation, the server processes/threads are appropriately synchronized to any shared data.” Applicants are not claiming multi-threading. More to the point, the discussion of a multi-threaded “single server process” and synchronizing shared data in such context does not disclose or suggest having content servers (two separate entities each running their own processes) share data and then having a mechanism (such as now positively recited) to inhibit content file sharing where the servers act upon purge instructions at different times (also neither disclosed nor suggested by Garrity et al.). Respectfully, it is error for the Examiner to simply lift a clause in the reference out of context and then apply that language to the claim.

A Notice of Allowance is respectfully requested.

Respectfully submitted,



Martin Korn  
Attorney for Applicant

Date: \_\_\_\_\_

4/15/05

LOCKE LIDDELL & SAPP LLP  
2200 Ross Avenue, Suite 2200  
Dallas, TX 75201  
214/740-8549 Telephone  
214/756-8549 Facsimile